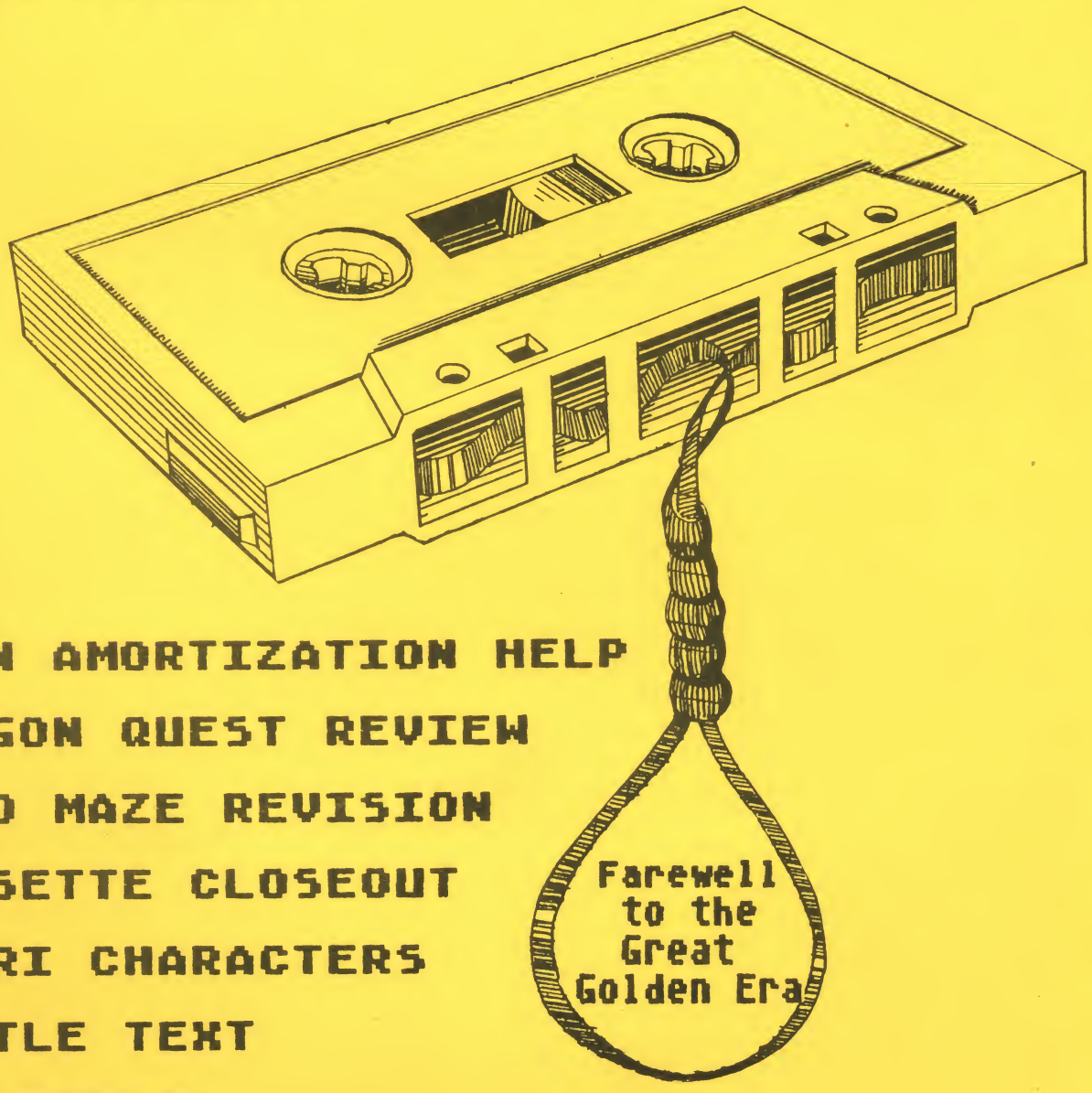
$1.50

March, 1985
Vol.5 No.3

M.A.C.E. JOURNAL

"Devoted Exclusively To The Atari Computer User"

IN THIS ISSUE...



Published by the Michigan Atari Computer Enthusiasts

FIRESIDE CHAT

Ok, all you Unknown Programmers! your time has come! At the April meeting we will hold our first MACE Amateur Night. Bring along your material (disk only) and show us what you've done. If anyone happens to own a gong, you might want to bring that, too! This event will be open to members only.

In March we will again have Dave Heinrich from The Family Computer Center available to answer your questions. Family is our local "Authorized Atari Service Center" and they were nice enough to agree to hold a Q & A session for the benefit of our members. So jot down your questions, whatever they may be, and bring 'em along to the March meeting.

I talked with Gary Nolan tonight. He's the President of Mil-Atari, a user group in Milwaukee, Wisconsin. They are attempting to organize 'Taricon 85! He was telling me about the contacts he's been trying to make. Seems the vendors don't want to respond until they see what Atari is going to do and it seems Atari's pledge to support user groups has slipped their minds. As usual, Atari hasn't returned their calls. I'll keep you posted as I learn more from Mil-Atari.

I hope everyone enjoyed Swap Night and I think we'll do it again soon. However, we will need a few volunteers to help organize it. If you're interested see any of the officers during the meeting. Advance registration and MACE membership numbers will be required.

Well, the Happy group purchase is over. The cut-off date was 01/15 and, unfortunately, orders received after that date will be returned. The OSS Tool Kits are here also and by the time you read this the Indus drives should have arrived. Watch for more group purchases in the future. One more word about past & future purchases! the cut off date is real. Orders are made the day after and we can't be responsible for late arrivals. Also, you MUST be a member!

'Till next month....

Kirk

UPCOMING MEETINGS

MARCH 19th

The March meeting is being billed as Expert Night, featuring Dave Heinrich of the Family Computer Center in Berkley. Dave will be here to answer your questions about the Atari, so come prepared. His presentation at the January meeting was well received; this should be another hit. We are also expecting a short demo of Omnimon.

APRIL 16th

As Kirk noted, April's meeting will give MACE's programmers a chance to show off their favorite programs (disk only, please). If anyone has a large gong available...

The schedule of meetings for the rest of the year is!

5/21	6/18	7/23	8/20
9/17	10/15	11/19	12/17

Tables will open at 7:00 pm. The actual meetings will start at approximately 7:30 pm.

^^*^*^*^*^*^*^*^*^*^*^*^*

The MACE Journal is published monthly by the Michigan Atari Computer Enthusiasts. Unless otherwise noted, material published in the Journal is in the public domain and may be reproduced for private or user group use providing proper credit is given.

Submissions to the Journal can be mailed to the PO Box, uploaded to the MACE BBSs or any officer's BBS, or uploaded directly to the editor at (313) 646-4455. Where possible, submissions should include a disk or tape file in AtariWriter or similar format and a working copy of the program. Specify format for screen dumps (AtariArtist, Koalapad, etc.). Authors whose submissions are published will receive a certificate good for a free disk or tape from the MACE library. Deadline for submissions is the first of each month.

LOAN AMORTIZATION

by Russell Crum

With long term loans today frequently involving more than one interest rate, I recently needed to reprogram an old amortization program which I had written some time ago. I think many readers will find this program useful with this feature.

The program allows up to four different interest rates in its calculation. The period of time that each rate is valid for as well as the total duration of the loan must be provided.

The output of the program is to a printer. The printout provides payment number, year of each payment, cumulative interest for the year and loan balance. This information is very useful in preparing tax estimates as well as evaluating alternatives on loan terms. Since it provides a complete amortization schedule, printer output is almost essential due to the length of the data output. A screen display could be used (with suitable program modifications) and the display stopped and started (press CTRL-1) to view the output at intervals.

The printer initialization in line 60 is to set the skip over perforation feature on the GEMINI 10X printer. If you have different printer codes, change this line accordingly.

Several lines in the program make use of coding available in OSS's BASIC XL and not in Atari BASIC. Line 30 sets up a format statement to use with the PRINT USING statements in lines 230 and 330. If you don't have BASIC XL, these lines will have to be altered by using appropriate syntax to position the output in columns. (See "Your Atari Computer" by Lon Poole, Chapter 4). Lines 100 and 170 also contain BASIC XL's prompted INPUT statement syntax (e.g. INPUT "prompt", variable). These are easily altered, if needed, to traditional Atari BASIC. Another BASIC XL feature can be seen in lines 400-430 in the concatenation of strings. This construction is A$=A$,"additional text" rather than Atari BASIC's A$(LEN(A$)+1)=" ". A final feature that I really like about BASIC XL: notice the "nice" line numbering. You don't really think I wrote this program without having to insert odd line numbers, do you?

The formulas used to calculate the payments, interest and balance may be found in numerous textbooks on this subject. This program has proven to be very useful and easy to use for my own applications. I think you will find it useful also.

$$$$$$$$$$$$$$$$$$$$$$$$$$$

```
10 REM LOAN AMORTIZATION
20 REM R. CRUM,OCT. 1984-USE BASIC XL
30 DIM A$(760),TC$(80),R(4,2):TC$="
       ####        ####.##
 #####.##                      ######.##"
40 GOTO 390
50 REM PRINT HEADER
60 OPEN #7,8,0,"P:":? #7;CHR$(27);CHR$
(82);CHR$(4);CHR$(27);CHR$(78);CHR$(4)
70 ? #7;TAB(38);Y:Y=Y+1:? #7;TAB(10);"
PMT NO";TAB(21);"PAYMENT";TAB(39);"CUM
 INTEREST/YEAR";TAB(66);"LOAN BALANCE"
80 ? #7
90 E=0:RETURN
100 ? CHR$(125):POSITION 10,0:? "LOAN
AMORTIZATION":INPUT "ENTER LOAN AMOUNT
 ",V
110 INPUT "ENTER NO. OF MONTHS OF LOAN
 ",N:F=-N:INPUT "ENTER MONTH NO. OF FI
RST PMT,e.g. JUNE-ENTER 6 ",A
120 FOR I=1 TO 4:? "ENTER INTEREST RAT
E FOR PERIOD ";I;" IN %/YR. ";:INPUT R
(I,1):R(I,1)=R(I,1)/100
130 ? "ENTER NO. OF MONTHS FOR RATE ";
I;:INPUT R(I,2):Z=0:FOR K=1 TO 4:Z=R(K
,2)+Z:IF Z=N THEN POP :GOTO 150
140 NEXT K:NEXT I
150 FOR J=2 TO 4:IF R(J,2)<>0 THEN R(J
,2)=R(J,2)+R(J-1,2):NEXT J:GOTO 170
160 POP
170 INPUT "ENTER YEAR OF FIRST PAYMENT
 ",Y
180 I=R(1,1):H=V*(I/12)/(1-(1+(I/12))^
F):H=INT(H*100+0.5)/100
190 ? CHR$(125):? "WHEN PRINTER READY,
 PRESS 'START'":POKE 53279,8
200 IF PEEK(53279)<>6 THEN GOTO 200
210 GOSUB 60
220 D=V:H=INT(H*100+0.5)/100:FOR B=1 T
```

```
0 N:C=D*(I/12):C=INT(C*100+0.5)/100:D=
D-(H-C):E=C+E
230 ? #7; USING TC$,B,H,E,D
240 IF D=0 THEN CLOSE #7:POP :END
250 IF A-INT(A/12)*12=0 THEN ? #7:GOSU
B 70
260 A=A+1
270 IF B=R(1,2) OR B=R(2,2) OR B=R(3,2
) THEN GOSUB 350
280 IF H>=D+D*(I/12) THEN POP :GOTO 31
0
290 NEXT B
300 END
310 REM LAST PAYMENT
320 C=D*(I/12):C=INT(C*100+0.5)/100:H=
D+C:E=C+E:B=B+1:D=D-(H-C)
330 ? #7; USING TC$,B,H,E,D:CLOSE #7
340 END
350 IF B>=R(1,2) AND B<R(2,2) THEN I=R
(2,1):GOTO 380
360 IF B>=R(2,2) AND B<R(3,2) THEN I=R
(3,1):GOTO 380
370 IF B>=R(3,2) AND B<R(4,2) THEN I=R
(4,1)
380 H=D*(I/12)/(1-(1+(I/12))^-(N-B)):H
=INT(H*100+0.5)/100:RETURN
390 A$="This program computes loan pay
ments and prints an amortization sched
ule.  It can handle four different"
400 A$=A$," interest rates for four di
fferent periods of the loan.  You will
 be asked for the loan amount,"
410 A$=A$," number of months for the l
oan, interest rate, number of months f
or each rate, number of the month"
420 A$=A$," payments begin, and year o
f first payment. The resulting table g
ives payment no., payment amount"
430 A$=A$,", cumulative interest/year,
 and loan balance for each year of the
 loan.  A printer is required!"
440 ? CHR$(125):FOR I=37 TO LEN(A$) ST
EP 37:FOR J=0 TO 37
450 IF A$(I-J,I-J)=CHR$(32) THEN 470
460 NEXT J
470 ? A$(I-36,I-J):I=I-J:IF (I+37)>LEN
(A$) THEN POP :GOTO 490
480 NEXT I
490 ? A$(I+1):? :? "PRESS 'START' TO C
ONTINUE":POKE 53279,8
500 IF PEEK(53279)<>6 THEN 500
510 GOTO 100
```

DRAGON QUEST

Reviewed by Cliff Siskin

Although Dragon Quest is a local product available from a Bloomfield Hills address, it is currently being advertised nationally in Antic magazine. The programs's small-time origins are evident in a variety of bugs, ranging from numerous misspellings in the manual and on the screen to contradictions between that manual and actual play (100% Wounds=Dead vs. 100% Wounds=Healthy) to screen displays that vanish before they can be read (the listing of your valuables available upon exiting the caves) to an "X" command that does not return you, as advertised, to an earlier part of the program. Surprisingly, none of these many, but essentially minor, errors ruin Dragon Quest; it is a challenging and enjoyable example of the Dungeons & Dragons-type "real time, animated graphics fantasy adventure."

Upon booting the program with BASIC, you enter a "Character Management Program" in which you build a character from core combinations of strength, dexterity, intelligence, gold, and weapons randomly generated by the computer. You select the race (human - for intelligence, dwarf - for strength and durability, or elf - for skill in archery), name, and sex of your adventurer. Programmer Matt Pritchard has, admirably, made sure that "if a character is female, she will be every bit as capable as a man but the scenario is changed so that Prince Anton," rather than Princess Anarea, must be rescued. A variety of characters can be stored on the program disk to be reviewed and used in different dungeon setups that can themselves be saved and reused in order to avoid constant remapping. Before leaving the character program for those dungeons, however, be sure to view the "Instructions" option. It tells you how to adjust the colors and brightness of the screen, program the joystick button for use of a particular weapon, press the proper keys for the remaining weapons, heal wounds, open treasure chests, look for secret doors, and even, if all hope is lost, commit suicide.

You may very well find yourself facing that final option many times and very quickly, for this is no easy quest. Your first task upon entering the "Game Play Program" is to negotiate with the honorable merchant Amad Ripuoff for weapons and healing potions. This process has been very cleverly programmed, requiring that you experiment and persevere with a variety of bargaining tactics in order to secure the most protection for your money. You will need all that you can get, for monsters lurk everywhere in the caves, giving you little rest. In fact, it is a good idea to save your character (and your dungeon) after the bargaining is completed so that if you meet your end quickly you will not immediately have to repeat your session with Mr. Ripuoff. Having taken that precaution, press the numeral "0" (not "O") to enter the first cave. There you must find the secret door leading back to the marketplace, for if you fail to do so the treasures that you are about to discover cannot be used to replenish your arsenal. Since your character can carry far fewer weapons than there are monsters, numerous returns are required. Returning after death is an easy but tedious process requiring pauses as your character is reloaded.

Movement within the caves, by keyboard or joystick, is sufficiently smooth and swift as are the battle movements. All actions are played out in primitive but clear graphics, consisting of the walls of 115 variably shaped rooms, mushroom-like treasure chests, and small figures representing you and the many monsters stalking the corridors. Descriptions of your status (wealth, weapons, wounds, and experience points earned by time in the caves and successful battles) are continuously displayed and updated. When monsters appear their names and strength levels are provided. The variety and complexity of your "negotiations" with these adversaries is as challenging and as pleasing as your encounters with Mr. Ripuoff; Pritchard's programming efforts clearly outclass those found in more well-known games, such as Morloc's Castle, by more prestigious companies, such as Epyx. Using the magic spells found in treasure chests is particularly entertaining. One of them, for example, entraps any monster, but one of your victims can still hurl a deadly fireball from inside his magical prison.

As of this writing, I am a rather accomplished warrior who has not yet mastered the art of "looking" for secret doors. Neither my male nor my female character has rescued its counterpart, but I look forward to doing so; Dragon Quest is a game in which frustration does not outweigh the lure of new surprises. If you are familiar with adventuring and know that you want to play this type of game, then Dragon Quest is a worthy choice even at a price ($24.95) that is high for this time of $100 computers.

Those MACE members who are not that familiar with adventuring or who are confused by the present proliferation of adventuring software should consult recent articles in Family Computing and Compute! magazines. In the January 1985 issue of the former, awards are given to the "Best Games of the Year" with the adventure genre subdivided into such current forms as "role-playing", "arcade/role-playing", "text-only" and "text/graphic." Two articles in the October 1984 issue of Compute! ("Is a Picture Worth a Thousand Words?" and "A Parser's Tale"), explain the differences between these types and how some of them work. Finally, any potential purchaser of one of these games should recognize that within the past few months an exciting alternative to playing somebody else's game has arisen: companies such as Codewriter, CBS Software, Electronic Arts, and Spinnaker have all issued programs that allow you to construct your own adventures. Although relatively primitive constructions sets have been previously available at the user-group level, these commercial efforts promise new levels of control over graphics and scenario complexity. For those attracted to the power of metaquesting - the quest to make a quest - Dragon Quest may still be a desirable purchase, not only for its own playability but also for the ideas for future play it may inspire.

Dragon Quest
Midwest Computing
4975 Brookdale
Bloomfield Hills, MI 48013
(313) 645-2140
Requires 32K and disk drive
$24.95.

MACE JOURNAL LISTING CONVENTIONS

To reduce our readers' eyestrain, we have adopted a special method for listing programs. Programs will be listed in 38 column format, and certain characters will be replaced by an abbreviated form of their function, printed within curly braces (see below). Any characters to be typed in inverse video will be underlined, and control characters will be represented by their respective letters within curly braces. If a character within braces is also underlined, toggle the inverse video on and then hold down the control key while typing the character.

This method may seem awkward at first, but you should quickly get used to it, and the listings will be much easier to read. The special characters which will be spelled out are as follows:

When you see:	You should type:
{CLEAR}	ESC SHIFT <
{UP}	ESC CTRL -
{DOWN}	ESC CTRL =
{LEFT}	ESC CTRL +
{RIGHT}	ESC CTRL *
{BACK S}	ESC DELETE
{DELETE}	ESC CTRL DELETE
{INSERT}	ESC CTRL INSERT
{DEL LINE}	ESC SHIFT DELETE
{INS LINE}	ESC SHIFT INSERT
{TAB}	ESC TAB
{CLR TAB}	ESC CTRL TAB
{SET TAB}	ESC SHIFT TAB
{BELL}	ESC CTRL 2
{ESC}	ESC ESC
{COMMA}	CTRL , (comma)
{PERIOD}	CTRL . (period)
{SEMI-COLON}	CTRL ; (semi-colon)
{SHIFT =}	SHIFT =

If you see:	Type:
{A}	CTRL A
A	INV. VIDEO A
{A}	INV. VIDEO CTRL A

SEVEN CITIES OF GOLD

Reviewed by Scott Barrett
Tri-Cities Atari Users Group

The date, April 4, 1492. The King has granted you 2,000 gold pieces for your voyage. You thank him for his generosity and move on. Your first stop is the Pub for a short rest and a drink, then home to say goodbye to your wife and have one more look at that map. Next, you go to the Outfitter to purchase your ships, crew, food, and trading goods. Finally it's off to the ship. You set sail and wonder if it's all worth it. Will you find gold and friendly natives, or fierce storms and danger?

That is the typical scenario of Electronic Arts' newest game, SEVEN CITIES OF GOLD. All of the action is shown on a scrolling hi-res graphics screen with lots of color. When you start to sail, the screen changes and you see a window in the center of the screen which shows where you are. Around the edges is information about food, men, the date, depth of the water or type of terrain and other related items.

When you encounter land, you send out an expedition. This can be of as many men as you like, carrying anything you choose from the ships. On these expeditions, you explore the land in search of gold and encounter natives of different tribes. You can trade with them, or you can conquer them. Some tribes are tougher than others, though. When you trade you can get gold, food, other natives to explore with you and maybe even set up a mission. You can even establish a fort in the New World. In this game you can use a historical map of the world as it was in 1492, or the computer will create a map for you using special geographical laws contained in the program.

All things considered, this game is lots of fun. There is only one thing I dislike about it. When you go into a native village, if you bump into the natives, you kill them. And sometimes it seems like they are trying to be killed because they walk right in front of you. There are lots of lands to explore out there, so get your crew together!

Every Product Discounted!

SAVE UP TO 50%

SECTOR INTERNATIONAL ONE

RELEASE THE MAGIC IN YOUR COMPUTER!

I ♥ MY ATARI

LEGEND

MPP

AMOD835D(D)	TERMINAL - 835/1030	10.00
AMOD835T(T)	TERMINAL - 835/1030	15.00
EDU6502 (D)	FOR NOVICE ASSEMBLY	10.00
EDU6502 (T)	FOR NOVICE ASSEMBLY	15.00
ETMODEM (D)	FOR 835/1030 MODEMS	15.00
MCMODEM (D)	TERMINAL FOR MCC-UM1	15.00
MPPDRIV (D)	MPP-1000C/E UTILITY	19.50

COMPUTER MAIL ORDER WORLDWIDE

37220 TRICIA DRIVE, STERLING HEIGHTS, MI. 48077

ORDER INFO. Phone: 313-978-2208 9am - 7pm. EST

ATARI®

free CATALOG

313-978-8087 MODEM 24 HOURS, MENU: P

GEMINI

ATARI COMPUTER HARDWARE PRODUCTS

CAT#		Your Price
ATARI	HARDWARE	
65XE	HOME COMPUTER 64K RAM SPRING	
130ST	HOME COMPUTER 128K RAM SUMMER	
520ST	HOME COMPUTER 512K RAM SUMMER	
600XL	HOME COMPUTER 16K RAM	69.94
800XL	HOME COMPUTER 64K RAM	111.94
1200XL	HOME COMPUTER 64K RAM	119.94
850	5 INTERFACES IN ONE UNIT	119.94
1010	PROGRAM RECORDER	39.94
1020	40 COLUMN PRINT/PLOTTER	69.94
1025	80 COLUMN MATRIX PRINTER	189.94
1027	80 COLUMN LETTER PRINTER	259.94
1050	DISK DRIVE WITH DOS 3.0	179.94
CX77	TOUCH TABLET	44.94
CX91	ATARI I/O CABLE 5 ft.	12.94
CX82	B & W MONITOR CABLE	26.67
CX85	NUMERICAL KEYPAD	28.57
CX86	PRINTER CABLE	26.67
CX87	INTERFACE/MODEM CABLE	26.67
CX88	INTERFACE/TERMINAL CABLE	26.67
CX89	COLOR MONITOR CABLE	26.67
CA014746	TV SWITCH BOX	3.15
CA014748	POWER ADAPTOR	14.40
CA016751	POWER ADAPTOR 830	15.38
CA017964	POWER ADAPTOR 400/800	15.38

ATARI ACCESSORIES

BX4201	ATARI 3-RING BINDER	4.58
BX4202	1025 RIBBON	2.97
BX4204	1020 PEN PACK (4) COLORS	2.90
BX4206	1020 PEN PACK (6) BLACK	2.90
BX4207	1020 PRINTER PAPER (2)	3.90
BX4208	ATARI LOGO MANUALS	26.50
BX4212	1027 INK ROLLER	3.88

HARDWARE FOR ATARI AND OTHER COMPUTERS

CAT#		Your Price
OTHER PRODUCTS FOR YOUR COMPUTER		
AT-13-6	I/O CABLE FOR ATARI 6ft.	15.00
AT-52-6	5200 15 PIN EXTENSION 6'	10.83
AT-5212	5200 15 PIN EXT CORD 12'	14.17
AT-F-13	FEMALE PIN ATARI IO PLUG	2.50
AT-M-13	MALE PIN ATARI IO SOCKET	2.50
C114016	ATARI CABLE 850 -IDS 6'	26.14
C115016	ATARI 850 TO EPSON 6'	25.50
C220016	ATARI 850 TO HAYES 6'	19.93
DE-9H	9 PIN HOOD	1.19
DE-9P	9 PIN PLUG (MALE)	1.59
DE-9S	9 PIN SOCKET (FEMALE)	1.93
DA-15H	15 PIN HOOD	1.36
DA-15P	15 PIN PLUG	2.27
DA-15S	15 PIN SOCKET	2.79
DB-25H	25 PIN HOOD	1.41
DB-25P	25 PIN PLUG	2.60
DB-25S	25 PIN SOCKET	3.45
DB3600	CENTRONICS 36 PIN MALE	6.60
DB3601	CENTRONICS 36 PIN FEMALE	7.07
HCROM	HAYES CRONOGRAPH	199.94
HSM300	HAYES SMARTMODEM 300	199.95
HSM1200	HAYES SMARTMODEM 1200A	479.95
LEG880	LEGEND 880 MATRIX PRINTR	249.94
MC24701	MALE - MALE 25 PIN 3'	17.07
MC25201	FEMALE-FEMALE 25 PIN 3'	17.07
MPP-1000E	MODEM WITH SMARTERM	109.97
MPP-1150	PRINTER INTERFACE	69.33
MPP-1150XL	FOR 1200XL ONLY	69.33
MPP-1152	2K BUFFER FOR MPP-1150	14.67
MPP=1156	SERIAL ADAPT. FOR 1150	41.25
MPP-PRT	MICRO-PRINT INTERFACE	59.98
SAFT200	200 WATT STANDBY POWER/A	388.24
SAFT400	400 WATT STANDBY POWER/A	528.24

GEMINI ELECTRONICS ACCESSORIES

CAT#		Your Price
Video Games/Computer Accessories		
VG150	JOYSTICK REPL. CABLE 56"	1.98
VG151	JOYSTICK EXT. CABLE 10'	3.98
VG152	JOYSTICK EXT. CABLE 20'	5.00
VG153	T.V. CABLE RCA PLUGS 12'	3.68
VG154	JOYSTICK 'Y' ADAPTER	3.85
VG155	JOYSTICK HANDLE BALLS	2.29
VG160	9 VOLT 500 mA AC ADAPTER	6.14
VG161	UNIV. 500 mA AC ADAPTER	7.68
VG165	ATARI 2600 GAME COVER	3.68
VG166	INTELLIVISION GAME COVER	3.68
VG167	COLECO GAME DUST COVER	3.68
VG168	UNIVERSAL DUST COVER	3.68
VG169	VAR. SPEED RAPID FIRE	7.68
VG170	ATARI TYPE JOYSTICK 800	5.97
VG171	DELUXE JOYSTICK 400/800	8.46
VG172	ATARI & C64 PADDLES PR.	12.23
VG173	TWO BUTTON JOYSTICK 2600	6.75
VG175	300 ohm GAME/TV SWITCH	2.54
VG176	300/75 ohm GAME/TV SW.	3.46
VG177	300 ohm 2 GAME/TV SWITCH	2.91
VG178	300/75 ohm 2 GAME/TV SW.	4.14
VG181	JOYSTICK REPL. PC BOARD	1.68
VG182	COMPACT JOYSTICK GEMSTIK	5.06
VG183	JOYSTICK WITH KEYPAD	13.66
VG186	INTELLIV. JOYSK. HANDLES	1.52
VG301	6 OUTLET STRIP W/PROTECT	24.54
VG302	6 OUTLET BOX W/PROTECTOR	13.83
VG303	COOLING FAN & 2 OUTLETS	34.15
VG304	VOLTAGE SPIKE SUPPRESSOR	3.83
VG305	JOYSK Y ADAPTER (APPLE)	9.98
VG306	JOYSK Y ADAPTER (T.I.)	5.37
VG310	5 PIN DIN-4 RCA PLUGS 6'	3.77
VG312	8 PIN DIN-3 RCA PLUGS 6'	6.14

1200XL WITH BASIC, 1020 PRINTER, 1050 DISK DRIVE $369.82 VALUE... 299.95

LOGO MAZE REVISED

by Bob Pettapiece

I really enjoyed the Logo Maze Construction Set by Ruth Gorishek in the December 1984 issue of the MACE Journal. However, like many others I decided to revise the program as I typed it in and ran it. Below are the revisions and a brief explanation in brackets why they were included. [Do not include the comments in the program!]

```
TO START
GO
MOVE
END
```

[I decided to use a more universal way to start the game.]

```
TO BUILD
CS HT
WHEN 0 [HT PU PLAY]
MAZE
END

TO FINISH
SETPN 0
LT 90 FD 20
PU HT
END
```

[TO BUILD was added to start the construction of the maze and to take the WHEN demon out of the TO FINISH procedure. This prevents a recursive procedure from playing the song twice.]

```
TO MOVE
ST
MAKE "CHAR RC
IF :CHAR = "- [PLOD]
IF :CHAR = "* [PLOD.R]
IF :CHAR = "+ [PLOD.L]
IF :CHAR = "= [PLOD.B]
IF :CHAR = "S [TS CT STOP]
MOVE
END

TO MAZE
FS PD SETPN 1
MAKE "CHAR RC
IF :CHAR = "- [AHEAD]
IF :CHAR = "+ [AHEAD.L]
IF :CHAR = "* [AHEAD.R]
IF :CHAR = "= [AHEAD.ERASE]
IF :CHAR = "F [FINISH SS STOP]
IF :CHAR = "L [AHEAD.LERASE]
IF :CHAR = "R [AHEAD.RERASE]
MAZE
END
```

[TO MOVE and TO MAZE were changed to use the arrow keys to move the turtle and to make the maze. The arrow keys are <u>not</u> used with the control key, as is normally done.]

```
TO PLAY
B A G A B PART B PART B REST
A PART A PART A REST
B PART B PART B REST
B A G A B PART B PART B REST
A PART A B A G G
WHEN 0 []
CT SS PR [PRESS "S" TO STOP]
END
```

[TO PLAY has not changed its tune, but the WHEN demon is turned off here (WHEN 0) so the song is not played twice. I also added a printed message to tell people what to do in order to stop.]

If you have already entered the program, these changes will be fairly quick to do using the Logo editor. Have FUN!!

MACE UNCLASSIFIEDS

Stan Rowland has gronked his copy of the Reporting Disk from an old Atari software package called The Bookkeeper Program. If anyone has a copy of this disk, please give him a call at (313) 243-3450.

COMPUTER RAFFLE

Win an Atari 400 with BASIC cartridge. Tickets will be on sale (50 cents each or 3 for a dollar) at the February and March meetings. The winning ticket will be drawn at the March meeting. You need not be present to win.

THE SHELL GAME
CRACKING ATARI LOGO

by Ann McBain Ezzell

If you have ever wanted to put text on your Logo screens, this month's column is for you. I'm not talking about dinky little Graphics 0 stuff at the bottom of the screen; I'm talking about big, bold Graphics 2 characters, with three colors, and normal and inverse video.

To understand how these procedures work, you have to know a little about how the Atari stores its character set, and how Logo handles the screen display. (You don't have to understand all this to use the procedures, but a little education never hurt anyone.) Atari characters are printed in an 8 x 8 matrix. The data for each character are stored in sets of 8 consecutive bytes starting at location 57344. (See page 143 of Ian Chadwick's Mapping the Atari for a more detailed description of the character set.)

The patterns of "on" bits in the 8 bytes for each character form the printed character on the screen. It is possible to copy this bit pattern into screen memory to draw characters on the screen in non-text graphics modes. It is easiest to do this in Graphics 8, where each bit in the screen memory controls a single pixel in the display, but it is possible to translate the bit patterns for other graphics modes.

Atari Logo uses Graphics 7 for its screen displays. There are 160 columns in a Graphics 7 display, with 40 bytes of screen memory used per line. 160 pixels per line divided by 40 bytes per line gives us 4 pixels per byte. With 8 bits in a byte, there are obviously 2 bits per pixel. Given 2 bits, we can represent 4 different binary numbers: 00 (=0), 01 (=1), 10 (=2) and 11 (=3). That is why there are four colors (counting the background) in a Graphics 7 display, and three pens with which to draw in Atari Logo.

If all of the bits in a byte are zero, the background color will be displayed. If the bit pairs are all 01s, four pixels in the color of pen 1 will appear. Bit pairs of 10 will produce the color of pen 2 and pairs of 11 will produce the color of pen 3. If you change the pen color with SETPC, of course the colors of the drawn lines will change. You can mix the bit pairs within a byte however you want to produce any desired pattern within the resolution limitations of Graphics 7.

To use this information to plot text on the Logo display screen, we must first determine the desired color and starting location of the text so that we will know what values to put into the screen memory and where to put them. Next we must determine the bit pattern to be reproduced and convert this into the proper values to put into screen memory. An 8 bit byte in the character set will translate into two bytes of screen memory, since each bit in the character set byte will become a pair of bits in screen memory. Once the translation has been done, the new values must be .DEPOSITed into the proper location in screen memory. A total of 16 bytes will be put in for each character to be printed. This may seem a little complex, but I think that it will all become clear as we work through the procedures.

The main procedure gets the information about pen color, starting location, and the text to be printed.

```
TO PRINT.TEXT
ST
CS
GET.PNUM
GET.XSTART
GET.YSTART
FIND.LOC :XSTART :YSTART
EACH.WORD GET.TEXT :LOC
END
```

The procedure GET.PNUM determines the color to be used to print the text. It assigns a value of one more than the pen number to a variable "PNUM which will later be used to determine the values to be .DEPOSITed into screen memory. It also checks to make sure that only numbers from 0 to 2 are entered.

```
TO GET.PNUM
TYPE [ENTER PEN NUMBER TO USE ( 0 - 2
) :]
MAKE "PNUM FIRST RL
```

```
   IF NOT NUMBERP :PNUM [PR [NUMBERS
ONLY, PLEASE] GET.PNUM]
   IF OR :PNUM < 0 :PNUM > 2 [PR [0 - 2
ONLY, PLEASE] GET.PNUM]
   MAKE "PNUM :PNUM + 1
   END
```

Next we need to determine the starting position of the text. The following procedures assign the column number to variable "XSTART and the row number to "YSTART. In GET.YSTART, if you will be using the full screen mode, change the prompt to (FS: 0 - 88).

```
   TO GET.XSTART
   TYPE [ENTER STARTING COLUMN ( 0 - 38
) :]
   MAKE "XSTART FIRST RL
   IF NOT NUMBERP :XSTART [PR
[NUMBERS ONLY, PLEASE] GET.XSTART]
   IF OR :XSTART < 0 :XSTART > 38 [PR [0 -
38 ONLY, PLEASE] GET.XSTART]
   END

   TO GET.YSTART
   TYPE [ENTER STARTING ROW ( 0S: 0 - 68
)]
   MAKE "YSTART FIRST RL
   IF NOT NUMBERP :YSTART [PR
[NUMBERS ONLY, PLEASE] GET.YSTART]
   IF OR :YSTART < 0 :YSTART > 68 [PR [0 -
68 ONLY, PLEASE] GET.YSTART]
   END
```

The next procedure calculates the starting position in screen memory from the values of "XSTART and "YSTART. 16384 is the address of the beginning of screen memory for a 64K configuration. Change it as needed if you have a 32K configuration (see the November Journal). Since each row contains 40 bytes, we take the starting address and add 40 times the number of rows plus the number of columns. If you start your text too close to the right side of the screen, it will wrap around, but the continuation will only be one row lower and will overwrite most of the previous text. Plan your screens accordingly.

```
   TO FIND.LOC :COL :ROW
   MAKE "LOC 16384 + ( 40 * :ROW ) + :COL
   END
```

Now that we know where to put the converted data, we have to get the text to be printed. The procedure EACH.WORD requires two inputs: a list (the text) and a memory location (the starting position). The input list comes as the output of the procedure GET.TEXT. EACH.WORD works on each word in the list, breaking it down into letters. LETTERS takes each letter and uses TEXT, CONVERT and GET.DATA to fetch the appropriate 8 bytes from the character set. POKE.DATA uses VALUE1 and VALUE2 to covert the two nybbles of each of those 8 bytes into the proper display bytes, then puts them into screen memory for LETTERS. Simple, right? Well, maybe not, but it works. Let's look at the procedures:

```
   TO EACH.WORD :LIST :MEM
   IF :LIST = [] [STOP]
   MAKE "NEWMEM :MEM + 2 * ( 1 + COUNT
FIRST :LIST )
   LETTERS FIRST :LIST :MEM
   EACH.WORD BUTFIRST :LIST :NEWMEM
   END
```

The variable "NEWMEM keeps track of the starting position for each word. Each character will require 2 bytes, and we need a space between words, so we add two times one more than the number of letters in the first word of the text to the original starting position. This becomes the starting position for the next word of text when EACH.WORD calls itself as it works through the text.

```
   TO GET.TEXT
   TYPE [ENTER TEXT TO BE PRINTED:]
   MAKE "TEXT RL
   IF :TEXT = [] [OP GET.TEXT]
   OP :TEXT
   END
```

This procedure simply outputs the text to EACH.WORD. Within EACH.WORD, LETTERS works on the individual words in the text.

```
   TO LETTERS :WORD :MEM
   IF 0 = COUNT :WORD [STOP]
   POKE.DATA :MEM TEXT FIRST :WORD
   LETTERS BUTFIRST :WORD :MEM + 2
   END
```

```
TO TEXT :LET
IF 127 < ASCII :LET [MAKE "INVERSE 1
MAKE "LET CHAR ASCII :LET - 128][MAKE
"INVERSE 0]
MAKE "ADDR 57344 + 8 * CONVERT ASCII
:LET
CLEAR.DATA
OP GET.DATA :ADDR
END
```

TEXT outputs to POKE.DATA the converted bytes to be put into screen memory. It first checks the ASCII value of each letter and sets the variable "INVERSE to 1 if the value is greater than 127 (indicating inverse video). In such a case, the variable "LET is converted to the corresponding normal video character. Otherwise, "INVERSE is set to 0. Next, CONVERT is used to change the ASCII value to the Internal Character Code (see page 55 in the BASIC Reference Manual).

```
TO CONVERT :VAL
IF :VAL < 32 [OP :VAL + 64]
IF :VAL < 96 [OP :VAL - 32]
OP :VAL
END
```

This converted code is used in TEXT to calculate "ADDR, the address of the appropriate 8 bytes within the character set. CLEAR.DATA clears out the variable "DATA, which will hold the 8 bytes from the character set.

```
TO CLEAR.DATA
MAKE "DATA []
END
```

GET.DATA outputs a list consisting of the 8 consecutive bytes which make up the desired character.

```
TO GET.DATA :LOC
IF 8 = COUNT :DATA [OP :DATA]
MAKE "DATA LPUT .EXAMINE :LOC :DATA
OP GET.DATA :LOC + 1
END
```

This list is output through TEXT to the procedure POKE.DATA, which uses VALUE1 and VALUE2 to convert each character set byte into two bytes for the screen display and .DEPOSITs them into the proper locations. The value of "MEM is increased by 40 each time POKE.DATA recycles to move the printing location down to the next line on the screen.

```
TO POKE.DATA :MEM :LIST
IF :LIST = [] [STOP]
.DEPOSIT :MEM VALUE1 FIRST :LIST
.DEPOSIT :MEM + 1 VALUE2 :LOW.NYBBLE
POKE.DATA :MEM + 40 BUTFIRST :LIST
END

TO VALUE1 :NUM
MAKE "VAL1 0
IF :INVERSE = 1 [MAKE "NUM 255 -
:NUM]
IF :NUM >127 [MAKE "VAL1 :VAL1 + 64
MAKE "NUM :NUM - 128]
IF :NUM > 63 [MAKE "VAL1 :VAL1 + 16
MAKE "NUM :NUM - 64]
IF :NUM > 31 [MAKE "VAL1 :VAL1 + 4
MAKE "NUM :NUM - 32]
IF :NUM > 15 [MAKE "VAL1 :VAL1 + 1
MAKE "NUM :NUM - 16]
MAKE "LOW.NYBBLE :NUM
OP :VAL * :PNUM
END

TO VALUE2 :NUM
MAKE "VAL2 0
IF :NUM > 7 [MAKE "VAL2 :VAL2 + 64
MAKE "NUM :NUM - 8]
IF :NUM > 3 [MAKE "VAL2 :VAL2 + 16
MAKE "NUM :NUM - 4]
IF :NUM > 1 [MAKE "VAL2 :VAL2 + 4
MAKE "NUM :NUM - 2]
IF :NUM > 0 [MAKE "VAL2 :VAL2 + 1]
OP :VAL2 * :PNUM
END
```

I am sure that there is a more elegant way to accomplish what is done by VALUE1 and VALUE2, and I would love to see it, but the brute force method works just fine, although a little slowly. Well, no one ever said that Logo was a speed demon. These two procedures break each byte from the character set into a high and low nybble and convert each into a byte to go into the screen memory. This is done by checking to see which bits are set (if bit 7 is set, the number will be greater than 127, since bit 7 carries a value of 128, and so on) and adding appropriate values to the number to be output.

These values are such that when their sum is multiplied by "PNUM, the resultant byte will produce the proper pattern and color on the screen. Perhaps an example is in order here. Let's say that the byte passed to VALUE1 is 36. This would mean that bits 2 and 5 are on; the others are off: 00100100. Let's also assume that this is part of a normal video character, so "INVERSE is not 1. Initially, "VAL1 is set to 0. VALUE1 will go through its statements until it comes to IF !NUM > 31, which will be true. "VAL1 will become 4 and "NUM will also be 4 (36 - 32). That makes "LOW.NYBBLE 4, too. Let's say that we are using pen 0, so "PNUM equals 1. The value output will thus be 4 (00000100). VALUE2 will start with 4 ("LOW.NYBBLE) as an input. "VAL2 will become 16, which will be the output (00010000).

Now let's look at the two output values in terms of bit pairs:

00 00 01 00 and 00 01 00 00

You can see that there will be 2 pixels in the background color (00), 1 in color 1 (pen 0), 2 more background, 1 in color 1, and 2 more background. This pattern is the same on/off pattern as originally found in the number 36.

If we were using pen 2, so that "PNUM equalled 3, the output values would be 12 and 48, with the following bit pair patterns:

00 00 11 00 and 00 11 00 00

The on/off pattern of pixels is the same here, but the color displayed will be color 3, which is pen 2. Pretty neat, huh?

One last item needs to be explained, then I will leave you to mull this over. In VALUE1, the value of "NUM is subtracted from 255 if the inverse video flag is set. Let's look at the bit patterns of two numbers, 19 and 236 (255 - 19):

19: 00010011
236: 11101100

See? One is the inverse of the other. The subtraction will make the characters printed on the screen appear in inverse video.

MACE GROUP PURCHASES

Here we go again...

Dataworld, the folks who have offered such great deals in the past, have done it once more. This month's specials are:

* MPP modem - $100
This autodial, autoanswer, direct-connect modem plugs into joystick port #2 on any Atari computer and does not require the 850 interface. It includes the Smart Terminal software cartridge, which supports the popular XMODEM transfer protocol.

* Indus GT disk drive - $230
This drive includes DOS XL, a word processor, database, and spreadsheet, plus a carrying case which doubles as storage for up to 80 diskettes. It supports single, double, and 1050 "density and a half". We need a minimum of 12 orders at this great low price.

* OSS language cartridges - $65 each
BASIC XL, MAC 65 assembler, and ACTION! cartridges - these are the actual languages, not the Tool Kits.

We also have an offer straight from the folks at Happy Computers:

* Happy 1050 Enhancement
Happy 810 Enhancement - $165 each
The 1050 version will not work on the newest 1050 drives, which have a WD2797 operating chip. Older drives have a WD2793 chip. This will enable your 1050 to run in true double density.

These offers are for MACE members only. To order any of the above items, send a certified check or money order to the MACE PO Box. Your order must be received by March 8th. Include your name, address, telephone number and MACE membership number. Be sure to specify exactly what you are ordering. If we do not receive 12 orders for the Indus drives, we will try to extend the offer, or we will return your check.

WHO

can you trust to provide

- Training
- Consultation
- Service
- Selection
- Price

The Family Computer Center

WE'VE JUST BOUGHT-OUT A WAREHOUSE OF ATARI, COMMODORE 64 AND VIC-20 SOFTWARE! TREMENDOUS SELECTION, PRICE!

543-0520

3895 W. 12 Mile Rd., Berkley
4 blocks east of Greenfield

SELL IT YOURSELF!

The USER-TO-USER Software Catalogue will offer low cost advertising for programmers who wish to sell directly to users. It will include an appendix of user groups. It will support Atari and several other personal computers. All types of programs are welcome, but should be priced between $5 and $20. The catalogue ($3-$4) will be advertised in national computer magazines. Advertising rates are $15 for up to 100 words to run for 3 months. Club listings will be $6 per quarter. The first issue will be released in the summer of 1985. Ads must be placed by April 30, with a special price before March 15. For more information, send a business sized SASE to:

USER-TO-USER
P.O. Box 2605
Eugene, OR 97402

LABEL PRINTER

Mike Mitchell is sysop of the Night Line BBS (11 pm - 8 am, (313) 525-5172) in Garden City, MI. This utility will print individual mailing labels and allows on-screen previewing.

```
0 GRAPHICS 2:POKE 710,0:POKE 752,1
11 POSITION 2,4:? #6;"mailing"
12 POSITION 12,4:? #6;"labels"
13 POSITION 0,7:? #6;"_________________
    "
15 ? :? "By Mike Mitchell * November 1
1, 1984"
16 FOR DE=1 TO 150:NEXT DE
17 FOR RB=1 TO 250:POKE 711,RB:SOUND 0
,RB,8,8:NEXT RB:SOUND 0,0,0,0
18 FOR SC=1 TO 4:PRINT :NEXT SC:? "
With on Screen 'Print Preview'":? :SOU
ND 0,75,10,8:FOR DE=1 TO 50:NEXT DE
19 SOUND 0,0,0,0:FOR DE=1 TO 700:NEXT
DE:GRAPHICS 0
20 ? "{CLEAR}":GOSUB 1000:POKE 764,60:
POKE 710,224:POKE 712,224:DIM N$(30),S
$(30),CS$(30)
24 POSITION 9,1:? " * MAILING LABELS *
 "
25 POSITION 1,3:? "_____________________
____________________"
26 POSITION 1,6:? "Name";:INPUT N$
27 POSITION 1,8:? "St.";:INPUT S$
28 POSITION 1,10:? "C,S & Zip";:INPUT
CS$
29 POKE 764,255
30 POKE 752,1:POSITION 2,15:? "Display
-> Screen  Printer  New  Quit"
31 R=PEEK(764):IF R<>62 THEN 33
32 GOTO 100
33 R=PEEK(764):IF R<>10 THEN 35
34 GOTO 200
35 R=PEEK(764):IF R<>35 THEN 37
36 GOTO 300
37 R=PEEK(764):IF R<>47 THEN 31
38 GOTO 400
100 POKE 764,255
101 POSITION 12,18:PRINT N$
102 POSITION 12,19:PRINT S$
103 POSITION 12,20:PRINT CS$
104 POSITION 2,19:? "Preview->"
105 GOTO 31
200 POKE 764,255
201 POSITION 12,22:? " Now Printing "
205 TRAP (500):LPRINT N$:LPRINT S$:LPR
INT CS$:LPRINT :LPRINT :LPRINT
215 POSITION 12,22:PRINT "{DELETE}{DEL
ETE}{DELETE}{DELETE}{DELETE}{DELETE}{D
ELETE}{DELETE}{DELETE}{DELETE}{DELETE}
{DELETE}{DELETE}{DELETE}{DELETE}":GOTO
 31
300 ? "{CLEAR}":? "{BELL}":POKE 764,25
5:POKE 752,0:GOTO 24
400 PRINT "{CLEAR}":PRINT "{BELL}":POK
E 764,124:POKE 752,0:POKE 710,148:NEW
:END
500 FOR RP=1 TO 3:POSITION 10,22:? "  T
urn on Printer!  ":SOUND 0,50,10,8:FOR
DE=1 TO 75:NEXT DE
501 SOUND 0,75,10,8:FOR DE=1 TO 75:NEX
T DE:NEXT RP
502 POSITION 10,22:? "{DELETE}{DELETE}
{DELETE}{DELETE}{DELETE}{DELETE}{DELET
E}{DELETE}{DELETE}{DELETE}{DELETE}{DEL
ETE}{DELETE}{DELETE}{DELETE}{DELETE}{D
ELETE}{DELETE}":SOUND 0,0,0,0:GOTO 31
1000 REM  Disable Break Key
1010 POKE 16,64:POKE 53774,64
1020 RETURN
```

PAGE ZERO

by Ann McBain Ezzell

If you read last month's column, you should remember that with one 8-bit byte you can express 256 different numbers, from 0 to 255. This month we will look at how these numbers can be used to communicate with our Ataris. Every letter, number or other character that is sent to or from your computer is transmitted as a single byte. So that one type of computer could communicate with another, or with different peripherals, an industry-wide uniform code was developed. It is the American Standard Code for Information Interchange, commonly shortened to ASCII (pronounced "ask-ee"). Atari has its own variation of this code, called Atari ASCII (ATASCII). The codes for upper and lower case letters, numbers, and most punctuation marks are the same as for "standard" ASCII, but special graphics and control codes are different.

Any good reference book for the Atari should have a chart of ATASCII values and their corresponding characters. Appendix C of the BASIC Reference Manual lists the values in decimal and hexadecimal form. You can also use the ASC and CHR$ commands in Atari BASIC to find the ATASCII value of a character or the character which corresponds to a given value. For example, if you type

```
PRINT ASC("A")
```

in immediate mode, the computer will respond with a 65. If you type

```
PRINT CHR$(65)
```

you will see the letter "A". ATASCII values for upper case letters range from 65 for "A" to 90 for "Z"; lower case values start at 97 for "a" and end with 122 for "z". The other values less than 128 are for numbers, punctuation marks, control and escape characters. Adding 128 to an ATASCII value generally causes the character to be printed in inverse video, but some codes have special functions. An example is 155, which is the code for End Of Line (RETURN) and is not printable.

If you take the time to play around with the CHR$ command for a while, putting in different values, you will probably come up with a few odd results. Try this:

```
PRINT CHR$(253)
```

Since the value is greater than 127, we would expect to see some sort of inverse video character. Instead, it rings a bell! Certain of the ATASCII values are associated with control codes such as BELL. These codes also have printable forms, but you have to tell the computer to treat them as display items rather than control codes. Location 766 ($2FE) is the display flag which the computer checks to determine how to treat these special codes. If you POKE 766 with a non-zero value, then ATASCII codes 27-31, 123-127, 187-191 and 251-255 will be printed as graphics characters. The codes will perform their normal display screen functions if the value in 766 is zero. If you have any of these characters in a string and want to print them, you will have to POKE 766 with a non-zero value. Hit BREAK or POKE 766,0 to regain the control functions.

So what use is all this? You can use the CHR$ function to store numeric data in string form and the ASC command to retrieve it. If you GET a byte from the keyboard, you will have the ATASCII value of the typed character. You can PUT ATASCII values to the screen (OPEN a channel to the S: device) to print characters. Strings can be alphabetized by comparing ATASCII values of the characters, since the values increase from A to Z.

What about the relationship between ATASCII codes and the Atari character set? The data for displaying each of the 128 normal video Atari characters is stored starting at location 57344. (Each character has eight bytes, which allows an 8 x 8 matrix of dots.) You might think that the characters would be stored in ATASCII order, but you would be wrong. If you look on page 55 of the BASIC Reference Manual or page 180 (Appendix 10) of Ian Chadwick's indispensable Mapping the Atari (you have bought your copy, haven't you?), you will find the Internal Character Code values listed. These values are primarily of interest if you are trying to find the 8 bytes of display data for a particular character (perhaps to

redefine part of a character set) or to assign colors to characters in text modes 1 and 2 (more about this in a later column).

The good folks at Atari weren't satisfied with two coding systems for their characters; there is a third set of codes used by the keyboard handler. These codes are listed on page 50 of the Operating System User's Manual. The internal hardware value for the last key pressed can be found in register 764. If no key has been pressed, the value is 255. The following line will let you check the value for a particular key if you don't have access to the OS manual:

```
10 PRINT PEEK(764): GOTO 10
```

Notice that the initial value is 255 and that once you press a key, the resultant value remains until you press another. SHIFT-A and CTRL-A will give different values from the A key alone. Pressing the CTRL key sets bit 7, which adds 128 to the value. Bit 6 is set by the SHIFT key, adding 64.

Whenever the keyboard is active, its handler monitors location 764 for a value other than 255. When it encounters one, it takes that key code and converts it to ATASCII form for additional processing, then restores 255 to await another keypress. You can use location 764 yourself to check for single inputs from the keyboard:

```
10 ? "PRESS ANY KEY TO CONTINUE"
20 IF PEEK(764)=255 THEN 20
30 POKE 764,255
```

The program will loop at line 20 until a key is pressed. If you do not POKE 255 back into 764 as in line 30, the character for the key pressed will be printed to the screen. You can also use this to check for keyboard input periodically within a loop while the computer is executing other commands. Another way to check for a keypress is to OPEN a channel to the keyboard handler and use the GET command, but the program will do nothing but wait for input from the keyboard.

```
10 OPEN #2,4,0,"K:"
20 GET #2,A
30 CLOSE #2
```

This will put the ATASCII value of the key pressed into the variable A, with nothing printed to the screen. If you write a program which requires single keyboard inputs at several places, you can put the above lines into a subroutine and GOSUB there every time you need to read the keyboard. You can then act on the value in A as needed.

I hope that you understand Atari's system of coding characters a little better as a result of reading this, and that what you have learned will be of some help the next time you sit down to write a program. Once again, if anyone has any suggestions for topics, please drop me a note at the MACE PO Box or give me a call at 646-4455.

CLOCK ROUTINE

This machine language routine will put a clock at the top of your screen. It will continue to run until you press SYSTEM RESET or write to page 6. You could include it in a game program to show elapsed time.

```
10 GRAPHICS 0:REM CLOCK BY Edward Chu
20 FOR Q=1535 TO 1676:READ A:POKE Q,A:
NEXT Q
30 ? "LEFT MARGIN";:INPUT A:POKE 0,A+6
4:? "SET TIME HR,MIN,SEC";:INPUT A,B,C
:X=A:GOSUB 60:POKE 203,X:X=B:GOSUB 60
40 POKE 252,X:X=C:GOSUB 60:POKE 253,X:
POKE 1,PEEK(106)-4:X=USR(1535)
50 ? "{CLEAR}":END
60 Y=INT(X/10)*16:X=X-INT(X/10)*10+Y:R
ETURN
100 DATA 104,169,11,141,36,2,169,6,141
,37,2,96,248,24,160,0,230,255,165,255,
201,6,144,56,24,132,255,165,254,105,1
110 DATA 133,254,201,16,144,43,24,132,
254,165,253,105,1,133,253,201,96,144,3
0,24,132,253,165,252,105,1,133,252
120 DATA 201,96,144,17,24,132,252,165,
203,105,1,133,203,201,19,144,4,24,200,
132,203,160,0,165,203,216,32,118,6
130 DATA 169,26,145,0,200,165,252,32,1
18,6,200,165,253,32,118,6,169,14,145,0
,200,165,254,133,204,32,130,6,76,98
140 DATA 228,133,204,74,74,74,74,24,10
5,16,145,0,200,165,204,41,15,24,105,16
,145,0,200,96
```

RITE WAY
COMPUTER CENTERS

AFTER INVENTORY SALE!

NEW ITEMS DAILY!

SALE ENDS FEB 28TH

16 DAYS ONLY

SLASHED PRICES!

MANY DEMOS MUST GO!

HUNDREDS OF OVERSTOCKED SOFTWARE TITLES

MONITORS
MODEMS
PRINTERS
DISK DRIVES

DONT MISS IT!

RITE WAY
SOFTWARE CLUB
~~$19.95~~ PER YEAR
NOW! $ 9.95

Hardware, Software, and Supplies For: Apple, Atari, Commodore And IBM

'THE' SOFTWARE RENTAL PLACE
RENTALS AS LOW AS $1.99 PER WEEK

ONE STORE ONLY!
DEARBORN
22027 MICHIGAN AVE.
DEARBORN, MI. 22027

RITE WAY
CASH REGISTERS • COMPUTERS

HOURS OPEN MONDAY THRU FRIDAY 10-7, SATURDAY 10-5

GREETINGS FROM YOUR EDITOR...

Hey - I got a letter! How about that? A new MACE member wrote in response to my plea in the February issue for suggestions for Journal material. So here are the questions and their answers; let this be an inspiration to the rest of you to dust off your Atariwriter cartridges and support your postal system.

1. Can anyone just sit down at one of the perimeter tables to sell something? Or do tables have to be reserved in advance?

The January meeting was a special Swap Night; table space was available only to members with reservations. Look for future Swap Night announcements in the Journal.

2. Is the Extended BASIC program in the February Journal available from the disk library? Are all Journal programs eventually available?

We will try to have all Journal programs available in the disk library. It would be nice to have them for sale the day that the Journal is distributed, but sometimes deadlines make that impossible. If you have a modem, you can download the programs from the MACE BBSs and several of the officers' boards.

3. How are submissions to the MACE libraries handled?

You can bring your programs on disk to the meetings and give them directly to Dave Zappa, Disk Librarian. You can also upload them to any of the MACE-affiliated boards (see inside back cover for a list). Be sure to leave a message for the sysop identifying your submission. Any public domain programs are welcome.

4. Does MACE have a library of back copies of magazines like Analog Computing and Antic?

No. You can order copies of most back issues from the magazines themselves.

5. Is there a printed roster of MACE members? It would be nice to have one, sorted by zip code, so people could find nearby Atarians to trade secrets with.

There have been several requests lately for a published membership list. Let's hear from the membership about this; would you want your name and phone number or address made available to other MACE members? Drop a note in the Suggestion Box or mail your opinion to the PO Box before March 1st and we'll publish the results in the next Journal. If too many members oppose a public membership list, we could publish names and numbers of those interested in contacting other members in a special column in the Journal.

6. At the meeting, the President said something about getting orders in for stuff apparently being ordered at special prices for MACE. Could these specials be noted in the Journal, so newcomers could find out about the latest deals?

MACE group purchases are announced in the Journal. Occasionally a short term offer will come in with a closing date such that it is not possible to advertise the offer in the Journal, but we try to avoid such situations. Remember that these special prices are available only to paid up members.

7. How is the master of the Journal printed? That is, which printer, interface, word processor, etc. It's a very good looking newsletter.

Thanks for the compliment. Articles are printed on MACE's trusty Atari 825 printer using the proportional font. Program listings and graphics are done on my Gemini-10. Both printers run through an Atari 850 interface (one at a time, of course). I use AtariWriter and its "print preview" option to get articles pared down to the right size. Listings are done with a program which I wrote that substitutes labels for special graphics characters ("UP", "CLEAR", and so on) and prints in a 38 column format. Most graphics and large text are printed with PRINTWIZ by Allen Macroware. Some of the designs are created with the Atari Touch Tablet.

Any more questions? Send them in!

SCIENTIFIC PROGRAMS

Book Review by Russell Crum

BASIC Programs for Scientists and Engineers by Alan R. Miller. Sybex, 1981. $16.95

This 320 page book is somewhat unusual in that it addresses scientific programs in BASIC. Such problems are most often found in a "scientific language" such as FORTRAN. The author developed these programs in 14 years of teaching engineering students. At the time of publication Mr. Miller was Professor of Metallurgy at the New Mexico Institute of Mining and Technology.

The text begins by discussing the limitations of some computers for scientific programs due to inaccuracies in their BASICs. There are some simple routines to test floating point and sine routines. I was pleased that my Atari 800XL came through with flying colors!

The other chapters address: mean and standard deviation, vector and matrix operations, simultaneous solution of linear equations, development of a curve-fitting program, sorting, general least-squares curve-fitting, solution of equations by Newton's method, numerical integration, nonlinear curve-fitting equations, and advanced applications which include the normal curve, the Gaussian error function, the gamma function and Bessel functions. In addition, two appendices present a summary of BASIC statements to help those unfamiliar with some of the syntax.

This is not just a cookbook of programs to be keyed in and run. Professor Miller presents the mathematical development of each program used to perform the operations. He frequently presents several programs to illustrate the evolution of the final form. In many cases more than one approach is presented due to potential difficulties with one version or another. A discussion of these difficulties is always given. Each chapter has several short problems to test the programs; answers are provided.

The BASIC used in the book is Microsoft BASIC-80, version 5. Translation is usually straightforward. BASIC-XL required no translation at all in many cases. For regular Atari BASIC, the majority of the translations involve prompted inputs (i.e. INPUT "prompt", variable), formatted print statements, and string dimensioning. A couple of the programs use the DEF FN statement, which can be replaced with a variable assignment and a GOSUB followed by variable reassignment.

The programs throughout the book follow a structured programming approach so that it is quite simple to combine them. For example, lines 1-499 are always main program, 500-880 are input routines, and so on. In fact, as the text proceeds, many programs make use of sections from earlier ones. These are indicated as such instead of being repeated at each occurrence. The author also clearly defines his variables in REM statements at the beginning of each program. These remarks provide two letter variable names as well as suggested "meaningful" names (e.g. D7 and DELTA). The listings use the two letter names. I found it easier to enter the program as listed, LIST it to disk and use AtariWriter's search and replace function to change to the alternative names. I prefer to use significant variable names rather than short forms which may be difficult to identify.

I have found working through the book to be quite interesting. It often brought back college memories of 20 years ago. How I wish I had had something like this and my computer then! My son has been able to use the simultaneous equation programs to check his Algebra II homework, also.

Who should buy this book? I would not recommend it to anyone who hasn't completed at least two years of algebra and trigonometry. A background in calculus would also be useful before tackling Professor Miller's explanations. The text is quite good, but it is not a math textbook. For those who would like to learn how to program some of these operations or who have a vocational application, I highly recommend the book. You will find that running some of these programs clearly demonstrates that the Atari is capable of much more than nice graphics and games!

CASSETTE CORNER

by Mike Landis

As everyone knows, disk drives have come way down in price. I've heard that you can find 1050's for under $150.00. Well, that's a long way from the $589.00 I paid for my 810 several years ago! Cassette drives were a hundred and some dollars; now you can buy a disk drive for that price! What I am getting at is, the cassette library is not in big demand. I have talked it over with the MACE Board members and we have agreed that we are no longer going to stock the cassette library. This does not mean they we are not going to handle cassettes; we will make them available on a special order basis only. By not keeping an inventory of cassettes, we will save MACE over $1000.00 in a year's time. Now, I know several ways we could spend $1000.00 to help MACE members! The disk and cassette library will publish a combined catalog. All you have to do when you want a program is order the cassette from the catalog and it will be mailed to you in about 10 days.

The next question is what are we going to do with all of the cassettes we now have in stock? Good question! We are going to sell them at $3.00 each, until we run out. Not only are we going to sell them that cheaply, but if you buy them in quantity we will discount them even more. Here is an example of quantity discounts:

4 Tapes for Only $10.00
9 Tapes for Only $20.00
14 Tapes for Only $30.00

To let you know how good these prices are, after these tapes are gone, the special order price will be $6.00. If you bought 14 special order tapes it would cost you $84.00, so you could save over $50.00 by buying in quantity now.

Now you are thinking, what's Mike going to do once he is no longer selling cassettes back in the corner? You will never guess! I will be selling blank disks! MACE has received a special discount on a large quantity of blank disks. We are going to give you a price that you will not believe! Ten blank disks with covers for only $10.00. (Can you believe that?)

Not only do I get to sell blank disks, I get to handle the raffle tickets! At our March meeting we are going to raffle an Atari 400 with a BASIC cartridge. Tickets are 50 cents each or 3 for $1.00. You can't beat that price! Even if you spend $5.00 and get 15 chances to win, isn't it worth it just helping MACE? Of course it is!

Well I hope I've answered all the questions. I will be in the corner providing cassettes, blank disks, raffle tickets and answering questions. See you there!

A HAPPY 1050!

by Kirk Revitzer

Well, I tried out the Happy Enhancement in my Atari 1050 and here's what I found: it works great! The Happy board requires that two chips be removed from the drive and it plugs into one of the sockets. The whole job takes about 20 minutes.

The software supplied was Version 6.0 which is compatible with the 1050 only. It does not support multi drive (Happy) use nor does it have features like Happy Compactor or Warp DOS. But, the enclosed warranty card says Version 7.0 will be shipped upon registration.

The Happy backup and special recovery features allowed me to back-up any disk in my library. No problems there.

Now for the best part! I booted up my version of "MYDOS" by Wordmark and inserted a double density disk in the drive. When I selected (A) for a directory the drive automatically switched densities and away we went. The Happy, I am told, supports single, DOS 3.0, and double density when used in the 1050 drive. I did not try a DOS 3.0 disk as I don't have any. The density capabilities are not mentioned in the documentation.

Well, anyhow, I'm happy!

WE'VE GONE SOLO!

OUR COMPUTER DEPT. NOW ONLY SUPPORTS

ATARI

THE BEST HOME-COMPUTER IN THE WORLD!

Join the major leagues!

Now forming a
Sunday evening
Microleague Baseball league.
Team Draft is 3/3/85.
Hurry! Space is limited.

Call for Details

New for your Atari

Basic XL Tool Kit	$30
Microleague Baseball	$29
Team Disks	$16
Print Shop	$35
Ghostbusters	$25
Syn-Stock	$30
Bank Street Music	$

SHOP US LAST!

Keep watching for all of Atari's newest computers & accessories

We'll have em' all!

Coming Attractions

Computer Video
22205 KELLY RD.
EAST DETROIT, MI
48021

313-445-BYTE

313-445-1313

M-F 11-9, Sat. 11-8, Sun. 12-5

Formerly Just Software

Diskbanks

HOLDS 50

$10.99

Disks

BOX OF 11

$10.99

After $2.00 rebate

CONTEST RESULTS

Well, the postman didn't exactly get a hernia carrying in the sacks of responses to our January contests. In fact, there were only two entries for Contest A, and none for Contest B. So much for group participation events in the Journal.

* CONTEST A *

Bob Retelle of Ypsilanti, MI recognized the data as a combination of musical notes and timing delays which played the Alphabet Song (or Twinkle, Twinkle Little Star). For each pair of numbers, the first was to be put into a SOUND statement:

```
SO. 0,A,10,8
```

and the second was to be used in a timing delay to sound the note:

```
FOR J=0 TO 10*B: NEXT J
```

Hugh McLean, sysop of Bunky's Board (546-3689), also recognized the data as music, but chose to use the timing delay numbers as color intensities to flash a little star on the screen while the music played. Close enough.

Both entrants will receive a certificate good for a free disk or tape from the MACE library. (I was going to choose randomly from among ALL the correct answers, but what the heck; they deserve something for caring enough to enter. Phooey on all the rest of you party poopers!)

* CONTEST B *

There was a hint in this problem - the part where the sorcerer put a "hex" on the data. If you take each number, convert it to its hexadecimal form, switch the digits, then convert it back to decimal, you will have the ATASCII values for the letters in the message. For example, take the first "hexed" value: 132. The hex form of 132 is $84. Reverse the digits to get $48, which is decimal 72, the ATASCII value of the letter "H". Get it? Now you can figure out the message yourself.

COUNTDOWN TIMER

by Jim Wilson

This subroutine will accurately define a delay in your program by using one of the countdown timers provided by the operating system. Using the system timers will ensure that the delays you design into your Atari BASIC program will remain constant when run from a faster executing language such as BASIC XL. This technique is also effective if you wish to speed up execution of the rest of your program with a BASIC COMPILER.

Poke a value into COUNTDOWN TIMER 3 (LOCATION 540,541) and it will count backwards to zero at the rate of 60 units (JIFFIES) per second. Enter the timer subroutine with the variable SECONDS set to the desired value (SECONDS=.5, SECONDS=10, etc.). SECONDS can be any value up to 1092 which will yield a delay of over 18 MINUTES.

The operating system uses timer 3 to OPEN the cassette recorder and to set the length of time to read and write tape headers. If you OPEN a channel to the cassette recorder while you are using this timer routine, the value in locations 540 and 541 will be lost. There are also timers available at locations 538/539, 542/543, and 544/545.

```
450 REM ***** Sample Execution *****
460 REM
470 FOR I=1 TO 10
480 ? I;" SECOND DELAY"
490 SECONDS=I
500 GOSUB 590
510 NEXT I
520 END
530 REM ***** Timer Subroutine *****
540 REM *                          *
550 REM *Set SECONDS prior to entry*
560 REM *                          *
570 REM ******************************
580 REM
590 TIMER=540:WAIT=SECONDS*60
600 WH=INT(WAIT/256):WL=WAIT-WH*256
610 POKE TIMER,WL:POKE TIMER+1,WH
620 IF PEEK(TIMER+1) THEN 620
630 IF PEEK(TIMER) THEN 630
640 RETURN
```

M. A. C. E.

MICHIGAN ATARI COMPUTER ENTHUSIASTS

P.O. Box 2785, Southfield, MI 48037

BULLETIN BOARDS: MACE 978-1685/MACE WEST 582-0657

MACE HOTLINE (Information - Voice Line) 882-7104

PRESIDENT
Kirk Revitzer
Detroit
882-9109
Compuserve: 70346,1642
BBS: 882-5909

VICE-PRESIDENT
Alva Thomas
Detroit
535-3748
Compuserve: 74065,334
BBS: 538-0197

TREASURER
Burt Gregory
Southfield
552-0273

PROGRAM COORDINATOR
Scott Garland
West Bloomfield
851-9453

CORRESPONDING SECRETARY
Sharie Middlebrook
Dearborn
581-5560
BBS: 582-0657

RECORDING SECRETARY
Dino Roggero
Redford
BBS: 531-1701

DISK LIBRARIAN
Dave Zappa
St. Clair Shores
773-8551
BBS: 771-4126

CASSETTE LIBRARIAN
Mike Landis
Clawson
589-1789
Compuserve: 72675,1023

MEMBERSHIP CHAIRMAN
Paul Wheeler
Detroit
538-3649

M.A.C.E. JOURNAL EDITOR
Ann McBain Ezzell
Birmingham
646-4455

NEXT MEETING: 3/19/85 7:00 PM
Southfield Pavilion--Ten & a Half Mile Road and Evergreen

M.A.C.E.
P.O Box 2785
Southfield, MI 48037

$20.00* FOR A ONE YEAR MEMBERSHIP
(*Payable to M.A.C.E.)

M.A.C.E. MEMBERSHIP APPLICATION

Name _____ Phone _____
Street _____
City _____ State _____ Zip Code _____
Company (if applicable) _____
System Description _____ Disk/Tape _____
Suggestions _____
I can help with . . . _____

( )New ( )Renewal
If a Renewal:
MACE #_____

Date _____ Coupon _____
Amount _____ Membership Card Number _____
☐ Cash Expires _____
☐ Check Number

MICHIGAN ATARI COMPUTER ENTHUSIASTS
P.O. BOX 2785
SOUTHFIELD, MICHIGAN 48037

IMPORTANT DATED MATERIAL

PLEASE DO NOT DELAY